MetaMap Data File Builder

Cliff Gay *

July 31, 2009

Abstract

The MetaMap algorithms address the task of automatically determining the concepts referred to in text. The default installation of MetaMap maps input text to concepts from the Unified Medical Language System (UMLS) Metathesaurus. The UMLS is focused on biomedical information sources. However, the MetaMap algorithms are not specific to the biomedical domain and can be generalized to any domain with adequate knowledge sources. The MetaMap Data File Builder enables this generalization.

1 Intended Use

If you want to use MetaMap to map text to the concepts in the entire UMLS Metathesaurus, then you will download one of the standard MetaMap data sets before running the installation program. If you have copyright issues or want to focus attention on selected Metathesaurus vocabularies, you will want to use MetamorphoSys to subset the Metathesaurus. If that is the case or you want to use MetaMap to map text to another knowledge source, then you will need the Data File Builder and should install it. This document will explain how to use the Data File Builder to create a custom data set.

2 The Problem

MetaMap uses a database that structures the information of the Metathesaurus to facilitate rapid access. While preparing the data files that are used to create some models of this database, the Data File Builder also filters out terms and strings that are unlikely to appear in normal English text and which thereby adversely affect the performance of MetaMap. There are word index files that speed the mapping from strings to concepts. There is also a file of MeSH treecodes that can be added to your output to facilitate post-MetaMap processing. And finally there are files containing the precomputed variants for strings in the Metathesaurus.

3 The Solution

There are three stages to preparing a new database for MetaMap:

1. Create a custom Metathesaurus
2. Generate MetaMap data files
3. Load the data files into a new database.

The following paragraphs introduce these stages, and later sections give detailed instructions.

3.1 Creating a Custom Metathesaurus

The source files for the data files must come from the UMLS Metathesaurus or must resemble the Metathesaurus files in format and semantics. Some hints and guidelines are provided in this section.

*revised by Willie Rogers

3.2 Generating MetaMap Data Files

This stage is the primary and time consuming part of processing the input Metathesaurus files into a form that supports rapid access to this data by MetaMap. The resulting organization and this pre-processing speeds up and improves the accuracy of MetaMap mappings. This processing is implemented by the Build Data Files module using seven scripts that use the GNU utilities and four specialized Java programs. [1]

3.3 Load Data Files

This final stage assembles the data files created in the previous stage into an MetaMap data set. This data set in then loaded into a new database. You will then be able to use this data for MetaMap processing.

The next section contains a discussion of how to interpret the instructions in this manual based on the computer platform you are using. That section is followed by a Background section which provides a historical perspective for Data File Builder processing. Next, the three processing stages defined above are discussed in detail. Finally, after some troubleshooting tips, the manual concludes with directions for using MetaMap with the new database.

4 Platform Differences

4.1 Solaris

You will need the GNU text utilities installed on a Solaris platform. Instructions for the download and installation can be found in "Installing Tools to Support Data File Builder" on the Installation page of the MetaMap web site. (http://mmtx.nlm.nih.gov/install.shtml#dfBuilder) You can run the commands shown in these instructions from most shell windows (sh, csh, tcsh, bash).

[1] *Check number of Java programs*

4.2 Linux

You will be able to run the commands and programs in these instructions from most shell windows (sh, csh, tcsh, bash).

5 Background

This section discusses the processing that goes into the generation of the necessary data files. There are several papers available on the Reference Page of the Semantic Knowledge Representation web site that explain portions of the Data File Building process.

Effective Mapping of Biomedical Text to the UMLS Metathesaurus: The MetaMap Program, 2001 This recent paper gives not only a good overview of the MetaMap processing available with MetaMap, but also discusses the data maintenance process and variant generation.

MetaMap Update Procedures, 2000 This paper describes the maintenance of MetaMap data files. It was this process that was adapted for use with MetaMap.

Filtering the UMLS Metathesaurus for MetaMap, 2001 A critical part of the MetaMap Data File generation is the filtering out of some strings from the source files. A brief description is provided below in the Filter section, but this paper explains manual filtering, the seven kinds of lexical filtering, filtering by Metathesaurus term type, and syntactic filtering.

MetaMap: Mapping Text to the UMLS Metathesaurus, 1996 Section 3 (Noun Phrase Variants, pp4-7) of this paper provides a good explanation of the role of variant generation in the MetaMap matching algorithm.

6 Creating a Custom Metathesaurus

There are two basic approaches to creating a custom Metathesaurus: (1) Use a subset of the UMLS Metathesaurus or (2) Use your own knowledge

sources. This section explains how to use either approach.

6.1 Create the Workspace

The Data File Builder expects the knowledge sources to be in a specific place. So you should create a workspace in which to put your knowledge sources.We suggest that you name your workspace and version to reflect the release of the UMLS Metathesaurus on which it is based. Specifically, we suggest you use the last two digits of the year version of the UMLS joined with a name meaningful to you by an underscore (e.g. "02_meshonly").

To create a workspace on Unix:

1. Decide what to call the version of data to be used. For this example we will use `04_custom`. This will be the workspace name.

2. Move to the directory mmtx just below the directory into which you installed MetaMap. This location may be defined by the environment variable BASEDIR.

   ```
   > cd $BASEDIR
   ```

3. Create a directory for the workspace

   ```
   > mkdir sourceData/04_custom
   ```

4. Create a directory to store the knowledge sources.

   ```
   > mkdir sourceData/04_custom/umls
   ```

6.2 Using a UMLS Metathesaurus Subset

The UMLS provides a tool, MetamorphoSys, to exclude and/or prioritize vocabularies so UMLS users can comply with licensing restrictions and exercise some control over naming of the Metathesaurus data. Complete documentation on its use can be found at `http://umlsinfo.nlm.nih.gov/mmsys.html` The UMLS Knowledge Sources manual (the green book) discusses the operation of MetamorphoSys in section 6. This information is available online at `http://www.nlm.nih.gov/research/umls/meta6.html` When preparing a subset with MetamorphoSys for use with MetaMap and the MetaMap Data Builder, you should designate your workspace as the location for the target files. To specify configuration for MetamorphoSys:

1. Move in your UMLS installation to the METAMSYS directory:

   ```
   > cd UMLS2004AC/META/METAMSYS/
   ```

2. Execute the script appropriate to your computer. For a Sun, type:

   ```
   > solaris_mmsys.sh
   ```

3. Select "Install UMLS."

4. For the Destination click on "Browse..."

5. Navigate the file browser until you can select the umls directory you created earlier.

6. Click the Open button. Your selected directory should appear in the Destination box. *Note: You may not need to install the other Knowledge Sources, so click on the Semantic Network and the SPECIALIST lexicon to deselect them.*

7. Select OK and then pick your desired subset.

8. When the UMLS MetamorphoSys Configuration tool comes up, select the Output Options Tab.

9. In the Select Output Format box, click on Browse button and change the format to Original Release Format.

Complete the configuration of MetamorphoSys and run it according its documentation. After it finishes, confirm that you now have in the umls directory of your workspace the files listed in the Required Files table of the next section.

File	When Required
MRCON	always
MRSAT	when treecode equivalent data is to be presented.
MRSO	always
MRSTY	always
SM.DB	always
semdef.txt	at least one semantic type must be present.

Table 1: Required Files

6.3 Using a New Knowledge Source

When you have a vocabulary that can be organized into concepts and would like to have those concepts be the target for the mapping of text by MetaMap, use the approach described in this section.

The goal is to create the files required to run the Data File Builder scripts in order to create a valid MetaMap database. Table 1 lists the files required. These files must be placed in the umls directory of your workspace.

Important Decisions: Correspondences from your knowledge source to Metathesaurus-like files need to be determined. The solutions will depend on the nature and structure of your knowledge source.

1. Identify a representation for synonymy. The fundamental basis of a thesaurus is the clustering of synonymous terms together as concepts. Find some relationship between the terms of your vocabulary that indicates synonymy. Some thesauri have *use* and *use for* relationships that identify a preferred term to which a group of other terms are related.

2. Identify a *preferred name* for each concept. Choose a term that best represents the concept. In the example above, the preferred term that only appears on the right of the *use* relation and only on the left of the *use for* relation is appropriate. This strategy of course will only work if the relations are well behaved. (We call the other terms in the concept *non-preferred*.)

3. Select a way to assign a unique identifier to each term in your knowledge source. It could be an identifier already assigned in your knowledge source or a sequential numbering of the terms in the input file.We will call this the *term id*. *Note: The MetaMap algorithms do not make significant use of the lexical structure imposed by the term identifiers (LUIs) of the UMLS Metathesaurus. The term id defined above is closer in meaning to the string unique identifiers (SUIs) of the Metathesaurus.*

4. Select a *concept id* for each concept (group of synonymous terms) in your knowledge source. It can be the term id of the preferred name.

5. We recommend that you attach a letter to the front of the required identifiers, following the example of the Metathesaurus. This will remind you of the meaning of the fields when you see the file.

One Approach: Here is the outline of a process you can use for creating the required files from your knowledge source. Let us suppose that your knowledge source is a thesaurus and you have made the decisions noted above. Then you:

- transform the thesaurus into a computationally friendly form;
- create the necessary identifiers for each term; and
- create the Metathesaurus files using the identified synonymy.

One approach for this is to take some text representation of your thesaurus and parse it into a line oriented, pipe separated text file. If your thesaurus is already in a database, you will not need to put it into that more computationally friendly form.

If you number the entries as you transform the file, then these numbers can be the basis for the identifiers.

Suggestions for filling in the columns of the required files are detailed in the following sections for each file.

MRCON The table below lists the fields and how they might be filled.

Field Name	Value	Source
CUI Concept Unique Identifier	D0000004	CUI letter + term id of preferred name (8 char)
Language of Term	ENG	Fixed
Term Status	P	Set to P for preferred name
	S	Set to S for non-preferred names
LUI Lexical Unique Identifier	M0000149	LUI letter + term id
String Type	PF	Fixed
SUI String Unique Identifier	T0000149	SUI letter + termid
String	Colleges	The term string
Least Restriction Level	0	Fixed

Table 2: MRCON Fields

MRSO The table below shows the fields and how to fill them. Please note that the example identifier values are for a non-preferred name–the LUI and SUI are different from the CUI. In our scheme the numeric part of all three would be the same in the entry for a preferred name.

Field Name	Value	Source
CUI	D0000004	Same as MRCON
LUI	M0000149	Same as MRCON
SUI	T0000149	Same as MRCON
SAB Source Abbreviation	CUSTOM	A name that identifies your thesaurus
Term Type	PT	if Preferred Term
	NP	if Non-Preferred
Source Identifier	0000004	term id
Restriction Level	0	Fixed

Table 3: MRSO Fields

MRSTY The table below lists the fields and how they might be filled. The semantic types and their ids used in this file must be defined in the semdef.txt file that was distributed with MetaMap. You will find this file in

Field Name	Value	Source
CUI	D0000004	Same as MRCON
TUI	T092	From semdef.txt
STY	Organization	From same row of semdef.txt

Table 4: MRSTY

MRSAT If you will not be using the treecode option of MetaMap, you do not have to supply a complete MRSAT file. It is used only to generate the treecode files that support an output option of MetaMap. Please check the introduction to Generating Treecode Files and the note about treecodes in the Data Set Creation and Load section below. Treecodes have been used in the past as a manageable representation of the parent/child relations of MeSH. Although their use is not part of the MetaMap algorithm, this information has been found useful for analysis after MetaMap processing. If you decide to generate "treecodes" for your custom dataset then use the table below as a guide for the MRSAT file. The indication of "not used" in the table, means that a constant value in this column will work as well as the value an UMLS MRCON file.

Field Name	Value	Source
CUI	D0000004	Same as MRCON
LUI	M0000149	Same as MRCON (not used)
SUI	T0000149	Same as MRCON (not used)
Source Identifier	MSH2004	Fixed
Attribute Name	MN Fixed	
SAB	CUSTOM	A name that identifies your thesaurus
Attribute Value	C15.378.71	Your treecode value

Table 5: MRSAT Fields

If you build your own MRSAT file, you must set

SAB to "MSH2004" or a modified equivalent with the year of the current UMLS release or the year of the UMLS on which your data is based.

If you decide not to generate "treecodes," then prepare a empty file and call it MRSAT.

semdef.txt If you will be using the semantic types from the UMLS you will not need to make any modifications. If you wish to add to or replace the semantic types from the UMLS Semantic Network you will need to modify the semdef.txt file. The verb—semdef.txt— is copied from the default data set into your new data set during data file generation. You can find it in `mmtx/data/<year>/mmtx/`.

If you do not care about semantic types, you can do the following:

To create a minimal semdef file:

1. Create a file called `semdef.txt`
2. Add a row with these values:

   ```
   objt|U001|Object
   ```

3. Save this file in a convenient location. You will move this file into you new data set after the other data files have been created.
4. Then assign this type to all your concepts in the MRSTY file.

SM.DB List of Semantically Related Terms (See http://www.nlm.nih.gov/research/umls/meta4.html) is in the file: `Install/<release>/LEX/LEX_DB/SM.DB`. Each row of the database has the following form.

```
TERM1|SCA1|TERM2|SCA2
```

Such a row indicates that TERM1 in syntactic category SCA1 is semantically related to TERM2 in syntactic category SCA2. Both terms are given in base form.

Examples:

```
alar|adj|wing|noun
ocular|adj|eye|noun
auditory area|noun|auditory cortex|noun
vomitive|noun|emetic|noun
vomitive|adj|emetic|adj
iridescent virus|noun|iridovirus|noun
typhloteritis|noun|cecitis|noun
```

An example knowledge source: Here is a sample entry from a thesaurus first shown as a normal document, then parsed into pipe-separated fields, and finally transformed into the required Metathesaurus files.

Thesaurus Entry:

```
Academic Institutions
 SN: All institutions of higher learning.
 UF: Colleges
     Community Colleges
     Universities
 RT: Administrators
     College Students
     Greek Societies
```

Parsed Thesaurus:

```
3|entry|Academic Institutions
Academic Institutions|SN|All institutions of higher learning.
Academic Institutions|UF|Colleges
Academic Institutions|UF|Community Colleges
Academic Institutions|UF|Universities
Academic Institutions|RT|Administrators
Academic Institutions|RT|College Students
Academic Institutions|RT|Greek Societies
. . .
149|entry|Colleges
Colleges|USE|Academic Institutions
Colleges|USE|Academic Institutions
. . .
155|entry|Community Colleges
Community Colleges|USE|Academic Institutions
```

MRCON entries:

```
D0000003|ENG|P|M0000003|PF|T0000003|Academic Institutions|0|
D0000003|ENG|S|M0000149|PF|T0000149|Colleges|0|
D0000003|ENG|S|M0000155|PF|T0000155|Community Colleges|0|
D0000003|ENG|S|M0000685|PF|T0000685|Universities|0|
```

MRSO entries:

```
D0000003|M0000003|T0000003|MMTX|PT|0000003|0|
D0000003|M0000149|T0000149|MMTX|NP|0000149|0|
D0000003|M0000155|T0000155|MMTX|NP|0000155|0|
D0000003|M0000685|T0000685|MMTX|NP|0000685|0|
```

MRSTY entries:

```
D0000003|U001|Object|
```

7 Generating the Data Files

There are five types of data used by MetaMap, and we generate the data files for each in separate directories. These types include the meta word index files, a treecode file, variants files, synonyms files, and

abbreviations and acronyms files. These data files are generated using a series of scripts because of the time required by some steps and to allow you to review some intermediate results. You will set up the workspace, move to each of the directories, and run one or more scripts to generate the data files for that type of data.

7.1 Setting Up a Workspace

We begin this stage by setting up the rest of the workspace started by your creation of the workspace directory and the `umls` directory within it.

To set up a new workspace:

1. Run this program:

   ```
   ${MMTX_PATH}/mmtx/bin/BuildDataFiles
   ```

2. Enter the name of your workspace at the prompt:

   ```
   > 04_custom
   ```

3. If the path shown is the location of the source data directory (umls), answer "Yes." If not, then enter the correct workspace name or move the data to the proper location. See Create the Workspace section above.

4. Enter lexicon year (4 digit year). Lexical data from the UMLS is used in the generation of variants and becomes part of your data files. Here you specify which version to use. (You usually will accept the default.)

   ```
   > 2004
   ```

If you get errors in this script, then your source directory does not have all the required files. See the table of required files in the section Using a New Knowledge Source above.

The workspace contains directories to hold the intermediate files and final output files as well as copies of the scripts you will execute.

7.2 Generating the Meta Word Index Files

The Meta Word Index files are the strings of the Metathesaurus (or your replacement knowledge source) indexed by the words they contain. These files are the majority of the MMTx data files. Your will generate them by executing fivescripts that will filter, and reorganize the raw data.

Workfiles The primary work of this first step is to strip the MRCON input file down to only the English strings.

To create the workfiles:

1. Move to the workspace directory, the one that contains the umls directory.

   ```
   > cd $MMTX/sourceData/04_custom
   ```

2. Enter the Meta Word Index directory:

   ```
   > cd 01metawordindex
   ```

3. Run 01CreateWorkFIles:

   ```
   > ./01CreateWorkFiles
   ```

The script will report some times and notes about what it is doing and remind you what to do next. It takes about 2 minutes to complete.

Suppress This step marks the terms that have been identified as problematic to MetaMap processing as suppressible synonyms so that they will be passed over during the filtering step. The small number of Metathesaurus strings that are problematic include numbers, single alphabetic characters, special cases, and ambiguities. There are two parts to this step; together they take about four minutes.

To run 02Suppress:

1. From the same directory in which you ran 0CreateWorkFiles (01metawordindex), enter

   ```
   > ./02Suppress
   ```

FilterPrep takes the concept file (mrcon.eng) and combines it with the sources file (MRSO) to create a file, mrconso.eng, with data from both that will be used in the filtering process. This step takes about 9 minutes. **To run FilterPrep:**

1. Still in 01metawordindex enter

```
> ./03FilterPrep
```

Filter The filter step uses studies previously performed on the full Metathesaurus dealing with how to

- process NEC/NOS,
- handle parentheticals, and
- handle possessives.

There are four forms of filtering:

- Manual Filtering – Those strings marked in the Suppress step.
- Lexical Filtering – Lexical filtering is the most benign type of filtering and consists of removing strings for a concept which are effectively the same as another string for the concept.
- Filtering by type –In addition to filtering out suppressible synonyms, terms are excluded based on their Term Type (TTY). The excluded types are generally abbreviatory, obsolete or have some kind of internal structure such as laboratory test descriptions in LOINC, one of the constituent Metathesaurus vocabularies.
- Syntactic filtering – The final kind of filtering is based on applying the parser to the Metathesaurus strings, themselves. Since normal MetaMap processing involves mapping the simple noun phrases found in text, it is highly unlikely that a complex Metathesaurus string will be part of a good mapping. Thus strings consisting of more than one simple phrase are filtered out. Because of their tractability, composite phrases (the ones containing wellbehaved prepositional phrases) are exempted from this filtering.

How much filtering is done depends on which model you pick. Because MetaMap is used both for highly focused, semantic processing as well as browsing, up to three data models differing in the degree of filtration can be created.

- Strict Model: All forms of filtering are applied. This view is most appropriate for semantic processing where the highest level of accuracy is needed.
- Moderate Model: Manual, lexical and type-based filtering, but not syntactic filtering, are used. This view is appropriate for term processing where input text should not be divided into simple phrases but considered as a whole.
- Relaxed Model: Only manual and lexical filtering are performed. This provides access to virtually all Metathesaurus strings and is appropriate for browsing.

To create a filtered model:

1. Verify you are still in `01metawordindex`
2. Run 04FilterRelaxed or 04FilterModerate or 04FilterStrict depending on which level(s) of filtering you desire. Note: If using the 04FilterStrict be sure that the SKR/MedPOST tagger is running. (To run the SKR/MedPOST tagger connect to public_mm directory and invoke bin/skrmedpostctl start—)

```
> ./04FilterStrict
```

This step runs a Java program to do the filtering of the mrconso.eng from the 03FilterPrep step. It then generates Filter/mrcon.eng.<model> from mrconso.eng.<model>. Finally, it creates a directory for the target model (either model.strict/, model.moderate/ or model.relaxed/) that contains a file, mrcon.filtered, which is linked to the appropriate Filter/ file. Generate MWI Files This final step for the Meta Word Index files produces the .txt files that will be loaded into the database. A Java program extracts word information from the filtered mrcon file and builds the necessary index files. This takes 12-15

minutes for each of the filter models you have created.
To generate MWI files:

1. While still in 01metawordindex enter

```
> ./05GenerateMWIFiles
```

If later you decide to generate additional models, you may run additional 04Filter scripts and then rerun this step. It will only generate the Meta Word Index files once for each model.

7.3 Generating the Treecode Files

The treecodes refer to the attribute of MeSH terms that encodes their hierarchical relationships to other terms. The treecode file does not support processing of text by MMTx; it is merely used to include treecode information as output.

Because the file depends on the Meta Word Index files which may occur in three models, there should be matching versions of the treecodes file. However, since MeSH in not strongly affected by the filtering process, only one version derived from the most inclusive model is computed. This step is the only one to use the UMLS MRSAT file. It also uses a special version of MRCON provided with your installation (mmtx/data/dfbuilder/MRCON.mesh).

To generate the treecodes file:

1. Move to the 02treecodes directory of your workspace. If you just finished generating the meta word index files, you can type:

```
> cd ../02treecodes
```

2. Run the script:

```
> ./01GenerateTreecodes
```

This only takes about 3 minutes and generates the file named treecodes.txt.

7.4 Generating the Variants Files

MetaMap variants are variants of words and terms that have been computed using a variety of variant generation methods. This mixture includes recursively defined derivational variants, recursively defined synonyms, acronyms and abbreviations, acronym and abbreviation expansions, spelling variants, and useful combinations of all the above methods. All inflectional variants of the resulting variants are also computed. A distance score for each variant from its generator is computed. Once the variants have been computed, the results are filtered so that only table entries with variants actually occurring in the Metathesaurus (the target of the variant generation) are kept.

The process of variant generation is based on a list ofwords in the knowledge sources. The words are the 'generators' for the variants. The words from the all_words table generated for the relaxed model and words from the Specialist Lexicon are used. For the latter data, the GenerateMMVariants program reads the file thatwas used to load the inflections table of the MMTx database.

Although the variants depend on the filtered model they are based on, normally variants are only computed for the most inclusive model available. So the script will use the relaxed model if available, the moderate model when the relaxed has not been generated, and the strict model when it is the only model available.

Variant Generation is a computationally intense process and the list of generators is typically 150,000 to 300,000 words. Therefore, this process typically takes less than four hours depending on the number of processors and speed of your system.

To generate variants:

1. Move to the 03Variants directory of your workspace. If you just finished generating the treecode file, you can type:

```
> cd ../03variants
```

2. Run the script:

```
> ./01GenerateVariants
```

Due to its long running time, external events may result in script termination before completion.Anormal completion can be recognized by the messages identifying the next step. If you encounter problems, consult the documentation of the GenerateMMVariants program at the MMTx web site: http://mmtx.nlm.nih.gov/docs.html. If you run those programs independently, be sure to include the –filterToTarget option.

7.5 Generating Synonyms

To generate synonyms:

1. Move to the 04synonyms directory of your workspace. If you just finished generating the variants files, you can type:

   ```
   > cd ../04synonyms
   ```

2. Run the script:

   ```
   > ./01GenerateSynonyms
   ```

7.6 Generating Acronyms and Abbreviations

To generate Acronyms and Abbreviations:

1. Move to the 05abbrAcronyms directory of your workspace. If you just finished generating the synonyms file, you can type:

   ```
   > cd ../05abbrAcronyms
   ```

2. Run the script:

   ```
   > ./01GenerateAbbrAcronyms
   ```

7.7 Replace Semantic Type Translations

If you modified a copy of the semdef.txt file or created your own, you should copy that file into your data set. A skeleton data set was created when you ran the BuildDataFiles script.

To replace semdef.txt:

1. Set your current working directory to the location of the your semdef.txt file.

   ```
   > cd <where it is>
   ```

2. Copy file into new data set:

   ```
   > cp semdef.txt $MMTX/data/04_custom/mmtx
   ```

8 Data Set Creation and Load

This final stage completes new data set and loads it into a new MySQL database. It assembles the various data files created by the previous stages into an MMTx data set labeled with the name you gave the workspace. This data set is then loaded into a new database. Afterwards you will be able to direct MMTx processing to this data just as you would to one of the standard datasets that you have downloaded.

To create a data set and load:

1. Run the LoadDataFiles program found in the $MMTX_PATH/mmtx/bin directory. If your current working directory was 04_custom you can type:

   ```
   > ../../bin/LoadDataFiles
   ```

   Note: On WIndows, do this in a Command Prompt window.

2. Enter the name of your workspace at the prompt:

   ```
   Enter workspace name (<year>_<source>)
   for model data [default is 04_custom]:

   04_custom
   ```

3. The program shows you the full path for the workspace. If it is correct, press return; if not, enter “no” and try again.

4. The program checks your workspace to see which models you have built. It will ask which ones you want to load. How long the load takes depends on the size of your source Metathesaurus. Our test subset takes less than half an hour.

The new dataset is found in mmtx/data/04_custom. A complete data set has at least two subdirectories containing the follocleanwing files, one subdirectory for the common files and one for the model specific files (in this case for the **strict** model):

If you get an error in this step (especially if it says, "File not found") it is likely that there was an earlier failure in one of the steps that was not noticed, or a step may have been skipped. Consult the table in the troubleshooting section below for how to handle this error.

This completes your use of the Data File Builder. Go to the Using MMTx section below for an explanation of how to point MMTx at the new database.

9 Troubleshooting

When a data file is missing. If a missing file is reported, you will need to go to the directory where it belongs and run or re-run the script that creates it. If you find intermediate files in that directory, check the output from the script carefully for error messages.We advise that you check that directory or the subdirectory with intermediate files for any empty files and remove them.

When an error is reported. There are many possibilities so we can only offer some general suggestions. The error may help you decide what to do. If not, clean up the working directory for the step that failed before trying it again

To clean up a working directory. Using the information from the Data File Builder Overview table, identify the directory used for the script that failed. Often it is the directory containing the script, but sometimes it is the directory containing the helper scripts or programs. From that working directory remove all files that are not input files. For 03variants, remove the subdirectories. (See the special troubleshooting note in the section Generating the Variants Files first.

When you need to re-generate MWI files. If there is an error or you modify the input data and rerun the Data File Builder scripts, the 05GenerateMWIFiles script will skip over those models for which it has already generated data files. To make it regenerate

DB/DB.normal.2008_level0.base:

config	metameshtc
config.01	meta_mesh_tc_opt.txt
config.02	nlsaa
config.common	nls_aa.txt
config.vars	nlsaau
cuisourceinfo	nls_aau.txt
cui_sourceinfo.txt	syns
cuisrc	syns.txt
cui_src.txt	vars
meshmh	varsan
mesh_mh_opt.txt	varsan.txt
meshtcrelaxed	varsanu
mesh_tc_relaxed.txt	varsanu.txt
meshtcstrict	vars.txt
mesh_tc_strict.txt	varsu
metamesh	varsu.txt
meta_mesh_opt.txt	

DB/DB.normal.2008_level0.strict:

all_words	first_words_counts.txt
all_words_counts	first_words_of_one
all_words_counts.txt	first_words_of_one.txt
all_words.txt	first_words_of_two
conceptcui	first_words_of_two.txt
concept_cui.txt	first_words.txt
conceptst	meshmh
concept_st.txt	meshtcrelaxed
config	meshtcstrict
cuiconcept	metamesh
cui_concept.txt	metameshtc
cuisourceinfo	nlsaa
cuisrc	nlsaau
cuist	sui_cui
cui_st.txt	sui_cui.txt
cui_sui.txt	sui_nmstr_str.txt
first_words	suistrings
first_wordsb	syns
first_wordsb_counts	vars
first_wordsb_counts.txt	varsan
first_wordsb.txt	varsanu
first_words_counts	varsu

Figure 1: Contents of a typical MetaMap dataset

the files follow these steps.

To regenerate files for the Strict model:

1. Go to the model directory.

   ```
   > cd 01metawordindex/model.strict
   ```

2. 2 Remove the sui_cui.txt file.

   ```
   > rm sui_cui.txt
   ```

3. Make the other files writeable.

   ```
   > chmod u+w *.*.*
   ```

4. Return to meta work index directory.

   ```
   > cd ../
   ```

5. Rerun the generate script.

   ```
   > ./05GenerateMWIFiles
   ```

When you need to start over. If there has been a problem with the input data and you need to rerun the Data File Builder scripts, you may run the whole process with a single command.

To run all Data File Builder steps at once:

1. Clean up the model directories. See previous section.

2. Start the run script:

   ```
   > $MMTX/config/rundatafiles.sh
   ```

3. You will be prompted to specify the dataset, it will probably already be the default.

4. You will also be ask which of the data models you would like generated. (Just running the strict model is the default.) Enter 'yes' for the ones you want.

10 Using MetaMap

This section gives a brief overview of using MetaMap with a custom data set. More complete information on using MetaMap and its options is available at the MetaMap web site (http://metamap.nlm.nih.gov).

The LoadDataFiles program created a new MetaMap configuration file that sets options to point MetaMap to the new database and the new variants table.

To run MetaMap:

1. Move to the top level directory in your MetaMap installation:

   ```
   > cd <parent directory>/public_mm
   ```

2. Using a input file of text to be processed such as the file used to test your installation, run MetaMap. (The command continuation characters 'áre used only for formatting here. You need to enter the entire command on one line.)

   ```
   > bin/metamap -Z 04_custom \
   resources/tes.txt custom.out
   ```

   Note: *The -A option (the default) specifies the strict model, use -B or -C to specify the moderate and relaxed methods.*

3. Examine the custom.out file to see the results with your new data set. The default installation generates tes.out with this input.

   ```
   > diff tes.out custom.out
   ```

   This command will probably return some differences because your newly generated custom data files are different from the data that came with the installation. Check to make sure that these differences are reasonable.